

24. (Once Amended) A computer-readable medium having a browser for executing HTML code received from a computer network having a computer, the browser comprising:

code programmed to analyze HTML code and identify scalar properties and expressions, each expression defining the value of a dynamic property and being a function of the scalar property;

code programmed to map a dependent/dependency relationship between the expression and the scalar property; and

code programmed to notice a change in a value of the scalar property and to execute the expression in response to the change in value of the scalar property.

Remarks

A. Objection to the Drawings

The office action objects to the drawings citing several transcription errors. The applicant has submitted proposed drawing corrections with changes marked in red ink. The applicant thanks the examiner for point out these errors and request approval of the drawing correction.

B. Objection to the Specification

The office action objects to the specification noting two typographical errors. The applicant has amended the specification to correct these errors and respectfully requests withdrawal of the objection to the specification.

C. Objection to the Claims

The office action objects to claim 3 stating that it does not limit the subject matter of its parent claim, claim 2. The applicant traverses this objection and notes that claim 2 sets forth the act of "changing the value of the second property" Claim 3 further limits this act by stating, "automatically changing" Therefore, the applicant respectfully submits that claim 3 further limits its parent claim and requests withdrawal of the objection to claim 3.

The office action objects to claim 17 noting that claim 17 appears to be dependent on claim 16. The applicant has amended claim 17 to correct its dependency and respectfully requests withdrawal of the objection to this claim.

The office action objects to claim 13 noting a typo that makes the second expression appear to be dependant on itself. The applicant has amended claim 13 to correct this typo and respectfully requests withdrawal of the objection to this claim.

D. Rejection under 35 U.S.C. § 112

Claims 4, 18, and 33 stand rejected as describing information that was not described in the specification in such a way as to enable one skilled in the art to make and use the invention. The applicant respectfully traverses this rejection.

A user interface is the portion of the program with which the user interacts. Furthermore, one skilled in the art understands that HTML code is a set of codes in a text file that instruct a computer how to format a document on the computer monitor. The document is typically a Web page or some other text file that is displayed for a user to view and interact with through any imbedded indexes or links. In other words, the web page or other text file generated using HTML code is a user interface.

Even if further explanation is needed, the specification states:

A Web page is typically created from an HTML document 106, which is a set of code in a text file format. The HTML document 106 controls the presentation and display of information on the computer 104. Examples of information that can be displayed on a Web page through the use of the HTML document 106 includes textual information, images, video, audio, and other executable programs.

Page 1, lines 16-20. This passage describes the generation of a user interface, even if it does not use that exact term. Furthermore, creating user interfaces with HTML is well known in the art and requires no further explanation other than the description of the invention set forth in the pending application.

Therefore, the applicant respectfully requests reconsideration and withdrawal of the pending rejection.

E. Rejection under 35 U.S.C. § 101

Claim 24 stands rejected as not containing patentable subject matter because the claimed invention does not tangibly embody the invention in a computer-readable medium and thus is not patentable subject matter. The applicant respectfully traverses this rejection, but has amended the claim to recite that the code is stored on a readable-computer medium. The applicant requests withdrawal of the pending rejection.

G. Rejection under 35 U.S.C. § 103

1. Claim 5

Claim 5 stands rejected as being obvious over Harris et al. in view of Garman. The applicants respectfully traverse this rejection.

Initially, the applicant notes that both Harris et al. and Garman are directed to programming in spreadsheets, which are application programs for calculating numerical values. Furthermore, spreadsheets are programmed by using formulas to establish relationships between cells. The teachings are limited to numerical calculations and fail to teach or suggest expressions written in a markup language that defines a dynamic property. In fact, these references fail to teach programming in any type of markup language or any other language that is coded in a text file. They also fail to teach or suggest programming formatting instructions, which programmers traditionally code using static values.

In sharp contrast to the cited reference, claim 5 sets forth a data structure in which an expression defining a dynamic property is written in a markup language, which is a set of code written in a text file that sets forth formatting instructions for a computer. This invention has significant advantages. For example, it permits describing relationships between objects, properties, and variables in terms of functions rather than an explicit sequence of steps. This advantage is significant because it simplifies programming in markup languages so that developers can program without knowledge of complex procedural programming techniques.

not
in
claims

Therefore, the applicant respectfully submits that no combination of the cited references set forth the claimed combination of elements. The applicant requests reconsideration and withdrawal of the pending rejection.

2. Claims 1-4, 19, 23, and 25

Claims 1-4, 19, 23, and 25 stand rejected as being obvious over Harris in view of Joseph Schmuller, Dynamic HTML: Master the Essentials, Chapter 7 (1998) ("Schmuller"). The applicant respectfully traverses this rejection.

Harris et al. is directed to programming in a spreadsheet, which is an independent application program for performing a specific task--calculating numerical values by using formulas to establish relationships between cells. The teachings are limited to an application program for numerical calculations and fail to teach or suggest a browser that can execute expressions that are a function of a scalar value. This application is executed independently of

any type of browser or any other program. Harris et al. also fails to disclose expressions that are a function of scalar properties and that are executed by a browser.

The copy of Schmuller provided in the office action merely discloses HTML code that is written using static values and does not disclose expressions that are functions of scalar values. One must use traditional procedural programming techniques. Furthermore, such HTML code is not an independent application program. It is executed by or through a browser.

There is no teaching or suggestion to combine Harris et al. and Schmuller to arrive at the claimed invention. In fact, one would not think to combine the two references for several reasons. For example, Schmuller discloses HTML code that uses static, procedural programming. Indeed, before the claimed invention HTML code was static and did not include the use of expressions that are functions of scalar values. A spreadsheet as described in Harris et al. is programmed by using formulas to relate one cell to another and is a fundamentally different style of programming. A person programming in HTML typically would not look to a spreadsheet for programming techniques. Independent spreadsheet applications and text files containing HTML codes as set forth in Schmuller employ two different types of programming techniques and serve two different purposes.

Another example of the differences between Harris et al. and Schmuller is that Harris et al. discloses an independent software application that operates independently and has its own programming rules. Schmuller discloses HTML code that uses procedural programming techniques and that must be parsed and have its syntax examined by a separate program (e.g., a browser). One programming with HTML code typically will not look to an independent spreadsheet application for suggestions on how to modify the HTML programming.

Therefore, the applicant respectfully submits that the claimed invention is patentably distinct from the cited references and requests withdrawal of the pending rejection.

D. Claims 6-18, 20-22, and 26-33

Claims 6-18, 20-22, and 26-33 stand rejected as being obvious over Harris et al. in view of Garman, and further in view of Schmuller. The applicant respectfully traverses these rejections.

1. The Cited References

Harris et al., Garman, and Schmuller were discussed above.

2. Claims 14-18, 26-28, and 29-33

Claims 14 (and hence dependent claims 15-18) and 29 (and hence dependent claims 30-33) set forth a method of and apparatus for processing dynamic properties in HTML code. Claims 14 and 29 set forth the acts of declaring or providing a value for one or more scalar properties; creating an expression that declares a value of a dynamic property, the dynamic property being a function of another property; and generating a dependency graph that maps the relationship between properties.

The copy of Schmuller provided in the office action merely discloses HTML code that is written using static values and does not disclose expressions that are functions of scalar values. One must use traditional procedural programming techniques. Furthermore, such HTML code is not an independent application program. It is executed by or through a browser.

There is no teaching or suggestion to combine Harris et al. and Schmuller to arrive at these claimed combinations. In fact, one would not think to combine Harris et al. and Garman with Schmuller for several reasons. For example, Schmuller discloses HTML code that uses static, procedural programming. Indeed, before the claimed invention HTML code was static and did not include the use of expressions that are functions of scalar values. A spreadsheet as described in Harris et al. and Garman is programmed by using formulas to relate one cell to another and is a fundamentally different style of programming. A person programming in HTML typically would not look to a spreadsheet for programming techniques. Independent spreadsheet applications and text files containing HTML codes as set forth in Schmuller employ two different types of programming techniques and serve two different purposes.

Another example of the differences between Harris et al., Garman, and Schmuller. Harris et al. and Garman disclose an independent software application that operated independently and has its own programming rules. Schmuller discloses HTML code that uses procedural programming techniques and that must be parsed and have its syntax examined by a separate program (e.g., a browser). One programming with HTML code typically will not look to an independent spreadsheet application for suggestions on how to modify the HTML programming.

Therefore, the applicant respectfully submits that the pending claims are patentably distinct from the cited references and requests withdrawal of the pending rejection.

3. Claims 6-13, 20-22, and 26-28

The applicant notes that claims 6-13 depend from independent claim 1, which is distinguishable from the cited references for the reasons stated above. Claims 20-22 depend from independent claim 19, which is distinguishable from the cited references for the reasons stated above. Claims 26-28 depend from independent claim 25, which is distinguishable from the cited references for the reasons stated above. Because these dependant claims incorporate all of the elements of these independent claims, the applicant respectfully submits that these claims are patentably distinct from the cited references as well.

The applicant respectfully requests reconsideration and withdrawal of the pending rejection of these claims.

Conclusion

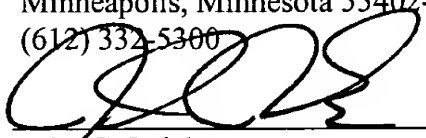
In light of the foregoing amendments and remarks, the applicant respectfully submits that the claims are in condition for allowance and requests advancement of the application toward issuance. The applicant also notes that there may be other arguments in support of patentability of the claims and reserves the right to raise any such argument in the future. Please call the undersigned attorney if there are any questions.

Respectfully submitted,

MERCHANT & GOULD P.C.
P.O. Box 2903
Minneapolis, Minnesota 55402-0903
(612) 332-5300

Date:

April 21, 2003



John C. Reich
Reg. No. 37,703



Version With Markings to Show Changes

In the Specification

The paragraph beginning on page 13, line 20, has been replaced with the following rewritten paragraph:

The dependency graph 401 forms a network of dependencies between the nodes. For example, the leaf node e 410 corresponds to a scalar property e, and the expression node c 408 corresponds to the variable or dynamic property c. Expression node c 408 depends on the leaf node e 410 and is its dependent node. Similarly, the leaf node e 410 is a dependency of the expression node c 408. Because the dependency is direct, expression node c 408 is a least-dependent node.

The paragraph beginning on page 14, line 9, has been replaced with the following rewritten paragraph:

The recalc engine 308 also creates a variable table 400, which is a lookup table of object/dispid pairs. In one possible embodiment, the lookup table is formed from a [dash] hash table of IDispatch objects. Each object entry is a list of dispid entries that correspond to a particular variable and point to the actual nodes in the dependency graph 401. The variable table 400 provides quick mapping between a variable and a node in the dependency graph 401. The variable table 400 also sets up and receives event notifications, which allows the object to listen for or notice events that affect a change in a property value. One possible mechanism for setting up and receiving event notifications is the OLE interface, IPropertyNotifySink.

In the Claims

Please amend claims 5, 6, 13, 17, and 24 as follows:

5. (Once Amended) A computer-readable medium having stored thereon a data structure, the data structure including a plurality of nodes, comprising:

at least one leaf node stored in memory, each leaf node containing a scalar property;
at least one expression node stored in memory, each expression node containing an

expression written in a markup language and defining a dynamic property, the expression being a function of the scalar property;

at least one pointer stored in memory, each pointer mapping a dependent/dependency relationship between the scalar properties and the expressions, wherein upon notification of a change in the value for one of the scalar properties, a browser executes the expressions dependent on the scalar property having a changed value.

6. (Once Amended) The computer-readable medium of claim 5 wherein[:]

the data structure is formed by a browser[; and

the expression is written in a markup language].

13. (Once Amended) The computer-readable medium of claim 12 wherein:

two pointers map the second expression to the first expression and a third expression, respectively, the second expression being dependent on the first and third expressions thereby being more dependent than the first and third expressions; and

the browser does not execute the second expression until the first and [second] third expressions are executed.

17. (Once Amended) The method of claim [10] 16 wherein:

the dependency graph includes a plurality of expression nodes mapped in network wherein the most-dependent expression node in the network has no dependent expression nodes and the least-dependent expression node depends directly on a scalar property; and

recalculating the values of expressions stored in nodes having a dirty state includes (a) executing the expression corresponding to each of the expression nodes in the network beginning with the least-dependent node and ending with the most-dependent node, and (b) assigning the value of each executed expression to the dynamic property corresponding to that executed expression.

24. (Once Amended) A computer-readable medium having a browser for executing HTML code received from a computer network having a computer, the browser comprising:

code programmed to analyze HTML code and identify scalar properties and expressions, each expression defining the value of a dynamic property and being a function of the scalar property;

code programmed to map a dependent/dependency relationship between the expression and the scalar property; and

code programmed to notice a change in a value of the scalar property and to execute the expression in response to the change in value of the scalar property.